

[INICIAR SESIÓN](#)[NUESTROS PLANES](#)[TODOS LOS CURSOS](#)[FORMACIONES](#)[CURSOS](#)[PARA EMPRESAS](#)[ARTÍCULOS DE TECNOLOGÍA > FRONT END](#)

# Cambiando CSS con JavaScript



Felipe Nascimento

22/12/2020

Actualmente, un operador tiene en su sistema web una tabla que consume algunos datos de una API. Entre ellos, las fechas en las que se debe realizar el mantenimiento de los vehículos. Lo que está sucediendo es que a los empleados les faltan algunas fechas de mantenimiento de vehículos. Eso ocurre porque dicen que la parte de la tabla encargada de mostrar las fechas no es visualmente clara. Tiene mucha información y la gente se pierde al leer la tabla.

Parte de la tabla de control utilizada por los empleados:

Conductor	Kilometraje	Modelo	Fabricante	Año	Última Revisión	Próxima Revisión
Pablo	100.504 km	24250	Volks	2012	2018-11-2	2018-12-30
Cristina	98.771 km	24250	Volks	2014	2018-11-2	2018-12-2
Octavio	151.123 km	24250	Volks	2013	2018-11-1	2018-12-1
Felipe	87.111 km	710	Mercedes	2011	2018-11-13	2018-12-13
Jose	130.574 km	710	Mercedes	2011	2018-11-23	2018-12-3
Maria	150.012 km	24250	Volks	2014	2018-11-12	2018-12-12
Fernanda	80.504 km	8150	Volks	2012	2018-11-15	2018-12-15

Después de tratar los datos de la API, se crea una tabla y los datos se organizan en HTML de la siguiente manera

```
<tbody id="tabla-control">
  <tr class="control">
    <td class="info-conductor">Pablo</td>
    <td class="info-kilometraje">100.504 km</td>
    <td class="info-modelo">24250</td>
    <td class="info-fabricante">Volks</td>
    <td class="info-ano">2012</td>
    <td class="info-ultimaRevision">25/11/2018</td>
    <td class="info-proximaRevision">25/12/2018</td>
  </tr>
</tbody>
```

¿Qué podemos hacer para mejorar la usabilidad del sistema?

## Cambiando el CSS

Sería interesante si tuviéramos algo que resalte visualmente las fechas, así que lo que vamos a hacer es crear un sistema de colores que represente los períodos de revisión.

Este sistema estará basado en tres colores:

- **Rojo** significará que está muy cerca o ha pasado el momento de la revisión.
- **Amarillo** significa que casi se te acaba el tiempo de revisión.
- **Verde** significa que todo está bien con la revisión, todavía queda mucho tiempo.
- Como hemos visto en el [artículo](#), pudimos realizar este tipo de cambio a través de los selectores. Con el mantenimiento y la legibilidad del código en mente, separaremos todas las responsabilidades en funciones. Comenzando por separar cada color en una función, comenzando con la función rojo.

Aquí recibe un parámetro td que representa el elemento que se modificará, en este caso, una celda de tabla.

```
function cambiaARojo(td) {
}
```

Ahora en el cuerpo de la función haremos cambios en el CSS, es decir, el estilo (style). En este caso, cambiaremos el color de fondo (backgroundColor):

```
function cambiaARojo(td) {  
    td.style.backgroundColor = "#FF0F0F";  
}
```

El valor #FF0F0F es la representación Hexadecimal de un tono del color **rojo**. Hagamos lo mismo con las funciones amarillo y verde:

```
function cambiaAAmarillo(td) {  
    td.style.backgroundColor = "#FFFA00";  
}  
  
function cambiaAVerde(td) {  
    td.style.backgroundColor = "#31FF4E";  
}
```

Genial, ahora que hemos creado las funciones, implementemos la lógica para cambiar los colores del campo según las fechas.

## Creando la lógica de los colores

Dado que las fechas dentro del campo son strings, capturaremos su contenido usando el `textContent`.

```
const fechaPagina = td.textContent;
```

Como no podremos hacer el cálculo correcto con las fechas, porque son strings, tendremos que encontrar una manera de convertir strings en fechas.

Convirtiendo una string en fecha Podemos instanciar el objeto `Date` y obtener la fecha actual

```
const hoy = new Date ();
```

Ahora, para convertir de string a fecha, instanciamos otro objeto `Date` con nombre `fecha` y pasamos como parámetro la fecha que está en el campo de **Próxima Revisión**.

```
const fecha = new Date( fechaPagina );
```

Ahora que hemos realizado la conversión, solo restar las fechas

```
const cantidadDeDias = calculoDeDias(hoy, fecha);
```

Ahora vamos a implementar una estructura de control if que se encargará de cambiar los colores del campo.

```
if (cantidadDeDias < 10){
    cambiaARojo(td);
}
else if (cantidadDeDias < 15){
    cambiaAAmarillo(td);
}
else
    cambiaAVerde(td);
}
```

Siguiendo nuestro estándar de separar las responsabilidades del código en funciones, creemos la función `cambiarColor()` que recibirá como parámetro un `td` que representa el campo de donde vamos a obtener la fecha.

```
function cambiarColor(td){
    const fechaPagina = td.textContent;
    const fecha = new Date( fechaPagina );
    const hoy = new Date();
    const cantidadDeDias = calculoDeDias(hoy, fecha);

    if (cantidadDeDias < 10){
        cambiaARojo(td);
    }
    else if (cantidadDeDias < 15){
        cambiaAAmarillo(td);
    }
    else
```

```
    cambiaAVerde(td);  
  }
```

## Calculando los días

La lógica de pasar de milisegundos a día es la siguiente: Primero dividimos por 1000 para obtener los segundos, luego dividimos por 3600 para obtener las horas y por fin dividimos por 24 y obtenemos el número de días.

```
const segundos = ( fechaActual - ultimaRevision ) / 1000 ;  
const horas = ( segundos / 3600 );  
const dias = ( horas / 24 );
```

Como puede suceder que no obtengamos un número entero, usaremos la función `Math.floor()` para redondear el número de días hacia abajo.

```
const cantidadDeDias = Math.floor(dias);
```

Ahora envolvemos la lógica en una función:

```
function calculoDeDias(fechaActual, ultimaRevision){  
  const segundos = ( fechaActual - ultimaRevision ) / 1000 ;  
  const horas = ( milisegundos / 3600 );  
  const dias = ( horas / 24 );  
  const cantidadDeDias = Math.floor(dias);  
  
  return cantidadDeDias;  
  
};
```

## Creando una verificación

Creemos una función llamada `verifica()` donde vamos a usar `forEach` que será el encargado de ejecutar la función `cambiaColor()` en cada elemento de nuestra array `ultimasRevisiones`

```
function verifica(ultimasRevisiones){
  ultimasRevisiones.forEach(td => {
    cambiaColor(td);
  });
}
```

## Dejando nuestra tabla actualizada

Para actualizar la tabla, usaremos un evento llamado `DOMContentLoaded`, que escuchará nuestra página y cada vez que se cargue, se actualizarán los colores de las alertas.

```
document.addEventListener("DOMContentLoaded", function(){
}
```

En el cuerpo de la función, usaremos el `querySelectorAll` para seleccionar todas las clases `.info-proximaRevision` para pasar los colores dentro de los campos, y luego llamar la función `verifica`, que aplicará la lógica de las fechas

```
const proximasRevisiones = document.querySelectorAll(".info-proximaRevision")
verifica( proximasRevisoes );
```



¡Tarea terminada!

Conductor	Kilometraje	Modelo	Fabricante	Año	Última Revisión	Próxima Revisión
Pablo	100.504 km	24250	Volks	2012	2018-11-2	2018-12-30
Cristina	98.771 km	24250	Volks	2014	2018-11-2	2018-12-2
Octavio	151.123 km	24250	Volks	2013	2018-11-1	2018-12-1
Felipe	87.111 km	710	Mercedes	2011	2018-11-13	2018-12-13
Jose	130.574 km	710	Mercedes	2011	2018-11-23	2018-12-3
Maria	150.012 km	24250	Volks	2014	2018-11-12	2018-12-12
Fernanda	80.504 km	8150	Volks	2012	2018-11-15	2018-12-15

## Estilos dinámicos con Javascript

Cambiar el color es solo una de las cosas que podemos hacer con Javascript, podemos capturar eventos, como clics en botones, scroll de páginas, entre otros para dinamizar las páginas.

Si está interesado en cómo funciona Javascript y cómo puede usarlo mejor, aquí en Alura tenemos [cursos de front-end](#). Ahí verás cómo programar en Javascript, usar expresiones regulares, entre otras cosas.

Puedes leer también:

- [Formatear números en JavaScript](#)
- [Capturando valor del input: introducción a los selectores](#)
- [Nombres de clases en CSS](#)

ARTÍCULOS DE TECNOLOGÍA > FRONT END

**En Alura encontrarás variados cursos sobre Front End.  
¡Comienza ahora!**

**SEMESTRAL**

**US\$49,90**

un solo pago de US\$49,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓

Estudia las 24 horas, los 7 días de la semana

- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 6 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**

[Paga en moneda local en los siguientes países](#)

**ANUAL**

**US\$79,90**

un solo pago de US\$79,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana





Foro y comunidad exclusiva para  
resolver tus dudas



Acceso a todo el contenido de la  
plataforma por 12 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**

Paga en moneda local en los siguientes países

Acceso a todos  
los cursos

Estudia las 24 horas,  
dónde y cuándo quieras

Nuevos cursos  
cada semana

## NAVEGACIÓN

PLANES  
INSTRUCTORES  
BLOG  
POLÍTICA DE PRIVACIDAD  
TÉRMINOS DE USO  
SOBRE NOSOTROS  
PREGUNTAS FRECUENTES

## ¡CONTÁCTANOS!

¡QUIERO ENTRAR EN CONTACTO!

## BLOG

PROGRAMACIÓN  
FRONT END  
DATA SCIENCE  
INNOVACIÓN Y GESTIÓN  
DEVOPS

AOVS Sistemas de Informática S.A  
CNPJ 05.555.382/0001-33

## SÍGUENOS EN NUESTRAS REDES SOCIALES



## ALIADOS



En Alura somos unas de las Scale-Ups seleccionadas por Endeavor, programa de aceleración de las empresas que más crecen en el país.



Fuimos unas de las 7 startups seleccionadas por Google For Startups en participar del programa Growth Academy en 2021

POWERED BY

## CURSOS

### Cursos de Programación

Lógica de Programación | Java

### Cursos de Front End

HTML y CSS | JavaScript | React

### Cursos de Data Science

Data Science | Machine Learning | Excel | Base de Datos | Data Visualization | Estadística

### Cursos de DevOps

Docker | Linux

### Cursos de Innovación y Gestión

Productividad y Calidad de Vida | Transformación Ágil | Marketing Analytics | Liderazgo y Gestión de Equipos | Startups y Emprendimiento